

---

**TECHNICKÁ UNIVERZITA V LIBERCI**  
Fakulta mechatroniky a mezioborových inženýrských  
studií

Studijní program: B2612 – Elektrotechnika a informatika  
Studijní obor: Elektronické informační a řídicí systémy

**Problémy J2ME z hlediska platformové  
nezávislosti**

**J2ME problems from the perspective of  
platform independence**

**Bakalářská práce**

Autor:	<b>Milan FUČÍK</b>
Vedoucí práce:	Ing. Miroslav HOLUBEC
Konzultant:	Ing. Miloš TUREK

**V Liberci 17. 5. 2006**

# TECHNICKÁ UNIVERZITA V LIBERCI

**Fakulta mechatroniky a mezioborových inženýrských studií**

Katedra: Katedra aplikované informatiky

Akademický rok: 2005/2006

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jméno a příjmení: Milan Fučík

studijní program: B 2612 – Elektrotechnika a informatika

obor: 2612R011 – Elektronické informační a řídicí systémy

Vedoucí katedry Vám ve smyslu zákona o vysokých školách č.111/1998  
Sb. určuje tuto bakalářskou práci:

Název tématu:

### **Problémy J2ME z hlediska platformové nezávislosti**

Zásady pro vypracování:

1. Popište architekturu programovacího jazyka Java určeného pro mobilní zařízení (J2ME), včetně stručné historie jazyka
2. Zdokumentujte co nejširší přehled typů zařízení, pro které je Java určena, včetně jejich technických parametrů
3. Konstatujte stávající řešení a problémy J2ME z hlediska platformové nezávislosti

4. Navrhněte možné úpravy vedoucí k odstranění těchto problémů při maximální snaze zachovat zpětnou kompatibilitu se současnou podobou J2ME

Rozsah grafických prací: dle potřeby dokumentace  
Rozsah průvodní zprávy: cca 40 stran

Seznam odborné literatury:

- [1] Juntao Y.M.: Enterprise J2ME: Developing Mobile Java Applications, 448 p., Prentice Hall PTR; 1st edition (October 20, 2003)
- [2] Topley K.: J2ME in a Nutshell, 478 p., O'Reilly Media, Inc.; 1 edition (March 20, 2002)
- [3] Muchow J.W.: Core J2ME Technology, 736 p., Prentice Hall PTR; 1st edition (December 21, 2001)
- [4] Sun Microsystems, Inc.: J2ME application program interface  
<http://java.sun.com/j2me/>

Vedoucí bakalářské práce: ing. Miroslav Holubec  
Konzultant: ing. Miloš Turek

Zadání bakalářské práce: **20.10. 2005**

Termín odevzdání bakalářské práce: **19. 5. 2006**

.....  
Vedoucí katedry

.....  
Děkan

V Liberci dne 20.10. 2005

## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum: 19. 5. 2005

Podpis: Milan Fučík

## **Poděkování**

Rád bych v úvodu poděkoval inženýru Miroslavu Holubcovi za vedení, metodické pokyny a cenné rady při tvorbě bakalářské práce.

Dále mé poděkování směřuje ke všem, kteří jakýmkoliv způsobem pomohli vzniku této práce.

V Liberci 19.5.2006

Milan Fučík

## **Anotace**

Bakalářská práce se věnuje problematice využití programovacího jazyka Java pro mobilní zařízení (J2ME) z hlediska platformové nezávislosti. Ta zažívá v současné době velice rychlý vývoj. V první části práce je uvedena stručná historie a popis architektury jazyka včetně základních principů jeho fungování. Další část je věnována dokumentaci co nejširšího počtu zařízení, jež používají výše zmíněnou technologii. Především jsem se zaměřil na nejprogresivněji rozvíjející se oblast mobilních telefonů. Ve dvou posledních částech jsem popsal stávající problémy platformové nezávislosti a navrhl možná řešení.

## **Annotation**

This bachelor's thesis is aimed at problems of the use of Java programming language in mobile devices considering platform independence. Nowadays, this area experiences extremely rapid development. The first part of the work briefly describes history and architecture of the language, including the basic principles. In the next part is examined documentation of broad variety of mobile devices. I put an emphasis on the most progressive domain, mobile phones. In the last two parts are described present problems in platform independence for which I suggested possible solutions.

## Obsah

Prohlášení	4
Poděkování	5
Anotace	6
Obsah	7
Seznam použitých symbolů a zkratek	9
1. Úvod	10
2. Architektura programovacího jazyka Java určeného pro mobilní zařízení	11
2.1 Stručná historie jazyka Java	11
2.2 Vlastnosti Javy	14
2.3 Vydání Javy	15
2.4 Architektura J2ME	18
2.5 Virtuální stroj	18
2.6 Konfigurace J2ME	19
2.7 CDC (Connected Device Configuration)	19
2.8 Profily CDC	20
2.8.1 Základní profil	20
2.8.2 RMI profil	21
2.8.3 Osobní profil	22
2.8.4 Herní profil	22
2.9 CLDC (Connected Limited Device Configuration)	22
2.10. Profily CLDC	22
2.10.1 Profil PDA	22
2.10.2 Profil MIDP	23
2.11 Architektura MIDP	23
2.11.1 MIDP 1.0	24
2.11.2 MIDP 2.0	24
2.12 Aplikace	26
2.12.1 Java Archive (JAR)	27
2.12.2 Java Application Descriptor (JAD)	27
2.13 Princip fungování J2ME	27
2.14 Vývojové prostředí	28
3. Zařízení používající J2ME	29
3.1 Set-top-box	29
3.2 Mobilní telefony	30
3.3 Smartphone	32
3.4 PDA	33
3.4.1 Palm	33
3.4.2 Pocket PC	34

4. Platformové problémy a jejich řešení	35
4.1 J2ME a bezpečnost	35
4.1.1 Koncept bezpečnosti	36
4.1.2 Nízkoúrovňová bezpečnost na úrovni virtuálního stroje	36
4.1.3 Bezpečnost na úrovni aplikace	36
4.1.4 "End-to-end" bezpečnost	37
4.1.5 Důvěryhodné a nedůvěryhodné sady midletů v MIDP 2.0	37
4.2 Hardwarové omezení J2ME	38
4.3 Softwarová omezení J2ME	39
4.4 Omezení virtuálního stroje pro CLDC	40
4.5 MIDP 3.0	42
5. Závěr	43
Seznam literatury	45
Seznam obrázků	46



## **Seznam použitých symbolů a zkratek**

J2SE	Java 2 Standard Edition
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
VM	Virtual Machine
API	Application Programming Interface
JVM	Java Virtual Machine
CDC	Connected Device Configuration
CLDC	Connected Limited Device Configuration
JNI	Java Native Interface
RMI	Remote Method Invocation
MIDP	Mobile Information Device
O-T-A	Over the Air
JAR	Java Archive
JAD	Java Application Descriptor
GUI	Graphic User Interface
MHP	Multimedia Home Platform
LBS	Location Based Services
PDA	Portable Digital Assistant
OLED	Organic light emitting diode

## **Ochranné registrované známky**

Java, Java ME™ jsou ochranné známky společnosti Sun Microsystems, Inc.  
Windows Mobile™ je ochranná známka společnosti Microsoft Corporation.  
Symbian OS™ je ochranná známka společnosti Symbian, Ltd.

## 1. Úvod

V posledních letech zažívá Java dynamický rozvoj, obzvláště její vydání určené pro malá mobilní zařízení. Trendem dnešní doby jsou stále menší a stále výkonnější platformy, jež jsou dostupné širokým vrstvám obyvatelstva. Patří sem PDA zařízení, mobily, spotřební elektronika a karetní systémy. Nejrozšířenější aplikací v této oblasti jsou bezesporu mobilní telefony. Dnešní mobilní telefony nabízejí možnosti, které bychom před deseti lety marně hledali u osobních počítačů. Výkon jejich procesorů je už dostatečně dimenzován na zvládnutí softwaru určeného pro vyšší platformy. Paměť v těchto zařízeních v dnešní době nepředstavuje problém, protože její kapacita již dosahuje řádově stovky megabytů a úroveň integrace je také na vysoké úrovni. Nesmíme opomenout také její cenovou dostupnost. Běžnějším se také stává připojení mobilních zařízení k Internetu a to buď prostřednictvím mobilního telefonu, nebo prostřednictvím WiFi.

S rostoucím výkonem a možnostmi zařízení stoupá také možnost vyvíjet pro dané zařízení aplikace. Převažují většinou samoúčelné aplikace jako hry atp. Aplikace se může spojit s jinou aplikací a vyměňovat si s ní informace. Tady už jsme schopni implementovat například vzdálený přístup do informačního systému atp.

Může to tedy vypadat, že Java vlastně žádné platformové problémy nemá. Ovšem není to tak docela pravda. V následujícím textu se budu snažit na ty nejdůležitější upozornit a navrhnou možná řešení.

## **2. Architektura programovacího jazyka Java určeného pro mobilní zařízení**

### **2.1 Stručná historie jazyka Java**

Za počátek vzniku Javy můžeme považovat podnět viceprezidenta společnosti Sun Microsystems Billa Joye k vytvoření nového jazyka, který by umožňoval stručný a efektivní zápis programů. V roce 1990 vznikl Green Team vedený Jamesem Goslingem. Tato malá vývojová skupina společnosti Sun si vytyčila za svůj cíl vytvoření systému pro domácí spotřebiče. Po dobu dvou let pracovala odděleně od zbytku světa na něčem, co by mohlo vnést nový život do světa informačních technologií.

Spotřební elektronika se vyznačuje podstatně delší životností (ať již morální či funkční) než současný software pro výpočetní techniku. Domácí spotřebiče mají omezenou velikost paměti a pomalé procesory. Cílem bylo vytvořit chytrou spotřební elektroniku, která by mohla být centrálně ovládána a programována dálkovým ovládáním. Za geniální se dá považovat myšlenka, že různá zařízení komunikují v heterogenním prostředí přímo mezi sebou. Green Team nejprve používal programovací jazyk C++. Ale ani on ani jeho modifikace však nebyly shledány vhodnými. Stejná situace nastala při pokusu o „oživení“ jazyka Pascal, jehož autorem je Niklaus Wirth (USCD Pascal). Nakonec James Gosling navrhl nový jazyk, který pojmenoval Oak (dub) podle stromu, jež rostl před okny jeho kanceláře. Za přímého předchůdce jazyka Oak můžeme považovat C++.

Koncová zařízení se osazují různými procesory podle požadovaného výkonu, a proto byl nově navržený jazyk interpretovaný, tj. nezávislý na konkrétním hardware. Požadavky na paměť pro základní interpret jsou 40 kB a pro standardní knihovny s podporou více vláken 175 kB. Nová technologie byla na světě, ale nedařilo se pro ni nalézt vhodné uplatnění. Do nelehké situace se nakonec vložil Bill Joy, spoluzakladatel společnosti Sun. Výsledkem byl úspěšný plán na její využití na poli Internetu. V té době se Internet stával stále populárnějším a Oak

představoval první možnost, jak oživit do té doby statické webové stránky. Malý kód, který se interpretoval prostřednictvím webového prohlížeče na straně klienta, byl nazván applet. Z bezpečnostních důvodů se applety spouštějí v tzv. „sandboxu“. To znamená, že mezi nimi a systémem vzniká bariéra, díky níž nemůžou applety přistupovat k souborovému systému a provádět další, potenciálně nebezpečné činnosti. Prvním programem pro spouštění appletů byl referenční prohlížeč Web Runner.

V roce 1995 bylo zjištěno, že jeden programovací jazyk s názvem Oak již existuje. Vlivem toho se pro programovací jazyk Oak začal užívat název Java a pro prohlížeč Web Runner název HotJava. Java byla poprvé představena na konferenci SunWorld 1995. Technologie byla licencována společností IBM, Microsoft, Symantec, Silicon Graphics, Inprise (Borland), Oracle, Toshiba či Novell.

Program v jazyce Java je interpretován prostřednictvím virtuálního počítače (Java Virtual Machine, JVM), díky čemuž je nezávislý na konkrétním hardwaru či operačním systému. Původně se jednalo o čistě interpretovaný jazyk, ale dnes z důvodu rychlosti převažuje JIT kompilace. Nejfrekventovanější části kódu (cykly) jsou za běhu (Just-In-Time) překládány do nativního kódu. Rychlost vykonávaného mezikódu (bytecode) ovšem nemůže dosahovat rychlosti nativního kódu.

Roku 1996 byl vydán první Java Development Kit (JDK 1.0), který obsahoval vše potřebné pro tvorbu appletů. Dnes je součástí SDK rovněž kompletní zdrojový kód platformy Java.

Společnost Sun definuje programovací jazyk Java jako jednoduchý, objektově orientovaný, distribuovaný, robustní, bezpečný, nezávislý na architektuře, portabilní, interpretovaný, vysoce výkonný a vícevláknový. Možnosti využití Javy se podstatně rozšířily s příchodem verze jazyka pro stolní počítače (od verze 1.2 označovaná jako J2SE). Zanedlouho se jazyk uchytil i v oblasti firemních serverů (J2EE) k vývoji aplikací, s kterými uživatelé komunikují prostřednictvím webových

prohlížečů (servlety, web service). Sun nechal Javu standardizovat, aby mohl i nadále sám rozšiřovat a řídit její vývoj prostřednictvím nově vzniklé Java Community Process (1998). Společnost Microsoft ovšem také prováděla svá vlastní rozšíření (nekompatibilní s verzí od společnosti Sun). Soudní spory vedly k odejmutí licence a následkem toho Microsoft přestal operační systém vybavovat aktuální verzí Java VM. Na poli webových prohlížečů se za několik let karty obrátily a většinový podíl převzal Internet Explorer společnosti Microsoft. Tyto události do značné míry omezily používání appletů a aplikací v Javě na stolních počítačích. Naproti tomu si na poli serverů dokázala J2EE vybojovat relativně pevnou pozici.

Posledním přírůstkem do rodiny je distribuce určená pro mobilní zařízení (J2ME). V podstatě se jazyk konečně začal používat k tomu, k čemu byl původně navržen. Díky J2ME dokáže programátor vytvořit vícevláknovou aplikaci pro mobilní telefon nebo PDA v nesrovnatelně kratším čase než dříve za použití assembleru konkrétního procesoru. J2ME se od J2SE neliší v jazyku, ale v dostupných aplikačních programových rozhraních (API). Například není možná práce s čísly v plovoucí desetinné čárce a s tím související matematické (tj. goniometrické) funkce.

Technologie Java se dnes používá v mnoha oblastech - od platebních karet (smart card) až po superpočítače. Dnešní domácí spotřebiče nejsou o mnoho chytřejší než byly v roce 1991, takže by se dalo říci, že původní výzkumný projekt společnosti Sun neuspěl. Namísto toho byl vytvořen nový objektově orientovaný jazyk pro programování, úzce svázaný s Internetem.

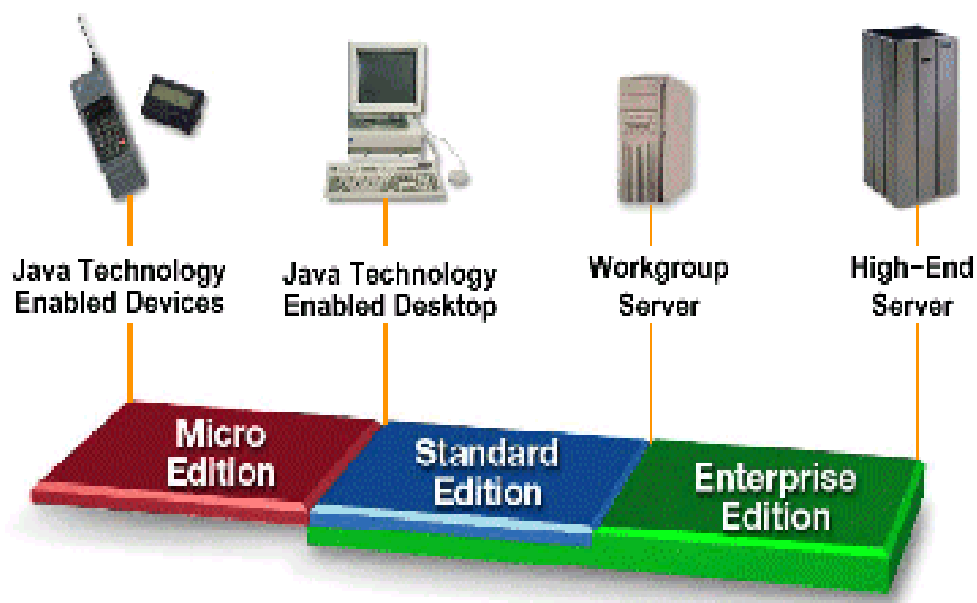
## 2.2 Vlastnosti Javy

- Java je objektově orientovaný jazyk, který vychází z jazyků C a C++,
- V Javě neexistuje přímý a nekontrolovaný přístup do paměti,
- Syntaxe je jednodušší a srozumitelnější,
- Je zaveden jednotný zápis pro přístup k objektům i knihovnám. Abecedy neanglických jazyků jsou podporovány pomocí šestnáctibitového znakového kódu Unicode,
- Prostřednictvím výjimek lze zachytit chyby a neočekávané stavy vyskytující se za běhu programu a umožnit jeho zotavení,
- Jazyk zahrnuje prostředky pro paralelní běh částí programu (multithreading) a jejich synchronizaci,
- Bezpečnost aplikací lze hlídat programovou kontrolou přístupu k objektům, vlastní kód programu je před spuštěním verifikován, podporovány jsou šifrovací standardy atd,
- Hlavním praktickým nedostatkem Javy je malá rychlost interpretovaných programů. To se týká zejména Internetových prohlížečů, které navíc mají mnohdy problémy se stabilitou appletů (ovšem nikoliv vinou Javy). Tuto vadu by měly v budoucnu odstranit vyvíjené JIT (Just-In-Time) kompilátory, které provádí překlad do strojového kódu procesoru.

## 2.3 Vydání Javy

Programovací jazyk Java je poskytován firmou Sun Microsystems ve třech vydáních:

- Java 2 Standard Edition (J2SE)
- Java 2 Enterprise Edition (J2EE)
- Java 2 Micro Edition (J2ME)



Copyright 1994-2007 Sun Microsystems, Inc.

Obrázek 1- Přehled použití *Java 2 Platform*

**J2SE** - zahrnuje kompilaci, nástroje, knihovny pro zápis a spouštění appletů a aplikací v Javě.

**Verze:**

- **JDK 1.0** (23. ledna 1996) – První release,
- **JDK 1.1** (19. února 1997) - významná rozšíření,
- **J2SE 1.2** (8. prosince 1998) — Kódové označení *Playground*. Tento a všechny následující releasy až do verze J2SE 5.0 byly přeznačeny,
- **Java 2** název verze "J2SE" (Java 2 Platform, Standard Edition) nahradil JDK z důvodu jednoznačného rozlišení základní platformy od J2EE (Java 2 Platform, Enterprise Edition) a J2ME (Java 2 Platform, Micro Edition),
- **J2SE 1.3** (8. května 2000) — Kódové označení *Kestrel*,
- **J2SE 1.4** (6. února 2002) — Kódové označení *Merlin*. Toto byl první release platformy Java vyvíjený komunitou Java Community Process pod označením JSR 59,
- **J2SE 5.0** (30. září 2004) — Kódové označení *Tiger*. (Původní značení bylo 1.5, které se dosud používá jako interní číslo verze),
- **Java SE 6** — Kódové označení *Mustang*. Tato verze je nyní vyvíjena pod JSR 270. Beta verze byla vypuštěna 15. února 2006.

**J2EE** - Java 2 Enterprise Edition - nadstavba nad J2SE pro rozsáhlé vícevrstvé podnikové aplikace. Jedná se o přístup (sadu pravidel, technologií, metod, doporučení), jak provádět design, vývoj, nasazení a provozování vícevrstvých aplikací pomocí jazyka Java formou několika základních komponent. Patří sem:

- Platforma J2EE - Standartizovaná platforma poskytující běhové prostředí pro J2EE aplikace,
- Soubor testů kompatibility J2EE - Sada testů ověřujících, zda je určitá platforma J2EE v souladu se standardem,



- Referenční implementace J2EE - Implementace J2EE určená pro prototypování J2EE aplikací; poskytuje definici J2EE platformy,
- J2EE BluePrints - množina nejlepších osvědčených a doporučených postupů při vývoji vícevrstevných aplikací.

### **Verze:**

- 5.0
  - zapracování nových rysů z J2SE 5.0 (anotace, ...)
  - EJB 3.0
- 1.4
  - Webové služby
  - J2EE Management
- 1.3
  - Enterprise Java Beans (EJB) 2.0
  - Základní podpora XML

**J2ME** - je vývojové prostředí pro psaní aplikací pro malá zařízení s podporou Javy, jako jsou například mobilní telefony nebo PDA (Palm OS). Avšak kvůli malému displeji, operační paměti a nedostatku místa, byla Java značně zjednodušena a upravena. Spousta knihoven vymizela nebo byla nahrazena změněnou a upravenou verzí. Například nebylo možné použít dosavadní systém oken, protože podpora jednotlivých oken je na mobilním telefonu skoro nepoužitelná. A tak přichází na řadu právě J2ME. Z toho důvodu má ve srovnání s J2SE vytvořen menší virtuální stroj a omezeny API funkce. J2ME jako taková, se dělí na konfigurace (configurations) a profily (profiles). Konfigurace jsou specifické pro určitou skupinu zařízení a určují vlastnosti Virtual Machine (VM) a Application Programming Interface (API), které toto zařízení musí minimálně implementovat. Skupiny zařízení jsou definovány na základě jejich systémových možností a výkonnosti, jako jsou například zařízení, která mají méně než 512 KB paměti. Pro tato systémově omezená zařízení je nutné definovat poněkud „ořezané“ VM, které budou implementovat pouze část funkcí určených pro standardní VM (viz. specifikace). Kromě toho,

že je omezena funkčnost VM, je zřejmé, že na dané platformě musí být omezena také nabídka API a obvykle se API soustředí na základní služby, které nabízí Java Standard Development Kit (java.lang.\*, java.io.\* apod.).

## 2.4 Architektura J2ME

			RMI	Osobní
Profil	MIDP	PDA	Základní	
Konfigurace	CLDC		CDC	
Virtuální stroj	KVM		CVM	
	Operační systém			

Obrázek 2 – architektura J2ME

## 2.5 Virtuální stroj

Java Virtual Machine (JVM) - sestává z *runtime systému*, tj. část realizující vazbu na hardware, a interpretu, který vykonává bytový kód.

## 2.6 Konfigurace J2ME

Konfigurace definují programové vybavení pro určité skupiny zařízení. Tyto skupiny jsou rozlišeny podle typu a velikosti paměti, typu a rychlosti procesoru a síťového připojení. Konfigurace zahrnuje virtuální stroj, knihovny, třídy a API funkce.

Rozlišují se dvě J2ME konfigurace:

- CDC (Connected Device Configuration )
- CLDC (Connected Limited Device Configuration )

## 2.7 CDC (Connected Device Configuration)

Tato konfigurace je zaměřena na zařízení s větším paměťovým prostorem (více než 2 MB), s 32 bitovým procesorem a síťovým připojením. Může se jednat o různé "vylepšené" spotřebiče (TV s internetem), herní konzole či jiná zařízení. Výkonná zařízení umožňují použití plného virtuálního stroje Javy, který zahrnuje:

- podporu plovoucích řádové čárky,
- JNI<sup>1</sup> (Java Native Interface),
- možnost definovat zavaděče tříd,
- serializaci objektů<sup>2</sup>,
- reflexi<sup>3</sup> .

## 2.8 Profily CDC

Profil doplňuje konfiguraci tak, že výsledná aplikace je lépe přizpůsobena vlastnostem dané skupiny zařízení.

---

<sup>1</sup> *JNI* - v některých případech je nutné volat kód, který není v jazyce Java. Například pokud je potřeba zpřístupnit specifické rysy operačního systému, k propojení se zvláštními hardwarovými zařízeními nebo opětovné použití již existujícího kódu, který není napsán v Javě. To vyžaduje podporu kompilátoru, virtuálního stroje a přídavné nástroje k připojení kódu Javy k jinému kódu. Standardní řešení k volání "ne-Javovského" kódu, který je poskytován JavaSoft, se nazývá *Java Native Interface*.

<sup>2</sup> *Serialize* - často by bylo vhodné objekty, které vznikají při běhu aplikace, uchovávat i po ukončení programu a obnovit jejich stav při dalším spuštění. Pro permanentní ukládání stavu objektů se vžil název serializace.

<sup>3</sup> *Reflexe* - umožňuje programu zkoumat sám sebe, měnit své vnitřní vlastnosti (např. je umožněno třídě získat jména všech svých částí).

**2.8.1 Základní profil** – konfigurace společně s profilem jsou základem architektury J2ME. Systémové vlastnosti a rozhraní je díky této architektuře skryto před aplikačním programátorem. Ten by měl svoji aplikaci zaměřit na využití profilu, který mu nabízí odpovídající API. Ve skutečnosti je aplikační vývojář od této vrstvy oddělen a ke konfiguraci jako takové by vůbec neměl přistupovat, pokud chce, aby jeho aplikace mohla být označena za „J2ME compliant“.

Z pohledu programátora je profil oním rozhraním, nad kterým staví svoje aplikace. Základní profil je vázán na CDC konfiguraci, ze které využívá dostupné Java balíky (viz výše). Důvodem této zvláštnosti je, že základní profil je u CDC zařízení pokládán za univerzální, a měl by proto být dostupný všem zařízením z této množiny.

Kromě toho je vidět, že základní profil ani konfigurace nedefinují žádný balík pro práci s uživatelským rozhraním. Důvodem toho je druh zařízení, pro které je CDC určeno (primárně pro autonomní zařízení jako jsou set-top boxy, TV apod.). Tato zařízení nejsou ve srovnání s osobními asistenty a PDA tak silně fixována na použití GUI, a proto v základním profilu nejsou GUI knihovny zahrnuty. Vlastní(Personal)Profil.

Specifikace pro Personal Profil byla vytvořena v rámci Java Community Processu. Přitom vznikl při veřejném dotazování dokument JSR-62 s kompletní specifikací. Jedná se o kompletní prostředí s plnou podporou AWT. Pro fungování tohoto profilu je nutné mít nainstalovaný základní profil, který je široce využíván (podobně je tomu u RMI profilu).

**2.8.2 RMI profil** (Remote Method Invocation) – profil RMI je určen pro platformy podporující CDC konfiguraci a pro svou práci vyžaduje přítomnost základního profilu, nad kterým je postaven. Základem RMI profilu je podpora pro tyto vlastnosti:

- plná podpora pro RMI sémantiku,
- „marshalování“ objektů,
- distribuované chování „garbage collectoru“ jak na klientu, tak na serveru,
- RMI registry a export vzdálených objektů,
- aktivační protokol na klientské straně.

Jedná se o nejdůležitější části RMI protokolu, které jsou zásadní pro jeho běh. Na druhou stranu bylo nutné některé části RMI odstranit, a to jak z důvodu systémových nároků, tak z důvodů předpokládané přenosové kapacity sítě. Vynechané jsou tyto funkce RMI protokolu.

- podpora pro průchod RMI přes proxy a firewall,
- RMI multiplexový protokol,
- podpora pro stub/skeleton model a stub/skeleton kompilátor,
- podpora pro „dřívější“ (deprecated) RMI rozhraní,
- Connected, Limited Device Configuration (CLDC).

**2.8.3 Osobní profil** – zahrnuje uživatelské rozhraní

**2.8.4 Herní profil** – podpora pro hry

## **2.9 CLDC (Connected Limited Device Configuration)**

Jedná se o nejmenší definovanou konfiguraci Javy. Uplatňuje se zejména v bezdrátových zařízeních s omezenými zdroji (mobilní telefony, pagery, PDA). CLDC je určen pro zařízení s paměťovým prostorem do 512 kB a 16-ti nebo 32 bitovým procesorem. CLDC definuje virtuální stroj Javy a minimální množinu knihoven a tříd.

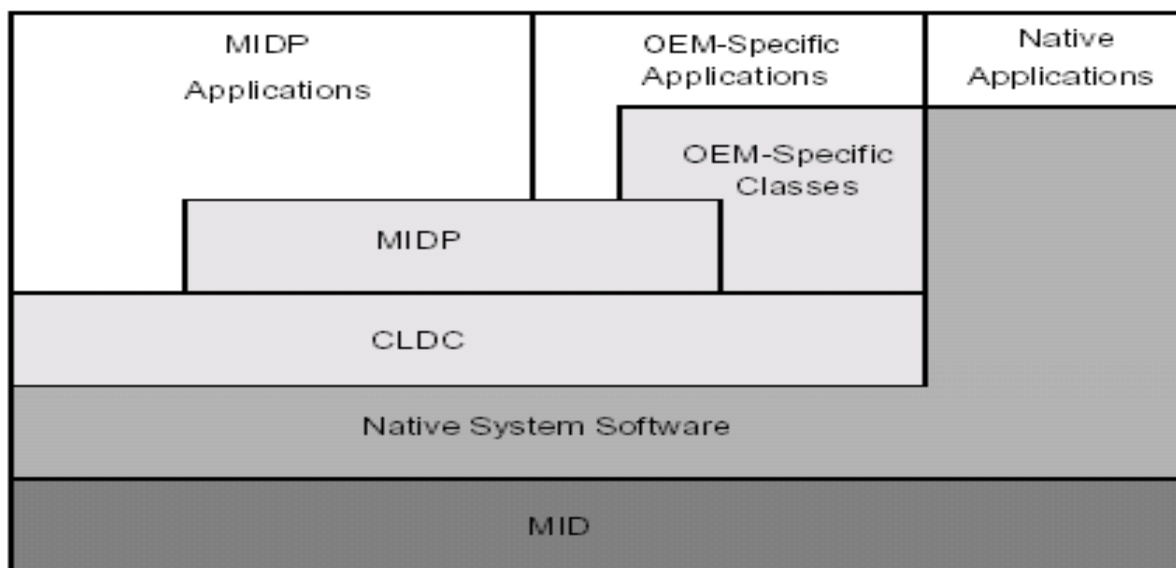
Základní požadavky CLDC jsou: 128 KB ROM, flash nebo jiné paměti pro trvalé uložení virtuálního stroje Javy, 32 KB paměti použitelné za běhu programu - pro dynamické požadavky Java aplikací (nahrávání tříd, alokace pro objekty a stack).

### **2.10. Profily CLDC**

**2.10.1 Profil PDA** – poskytuje programové prostředky uživatelského rozhraní a ukládání dat v zařízeních PDA.

**2.10.2 Profil MIDP** - Protože CLDC je určena pro velmi širokou a rozmanitou škálu zařízení, existuje zde velmi vysoký potenciál pro vytvoření specifických profilů pro každou kategorii zvlášť. Dnes asi nejpopulárnějším a nejznámějším je Mobile Information Device Profile (MIDP), který je určen pro mobilní telefony a pagery. Tento profil je postaven nad CLDC, které tvoří vrstvu pod ním.

## 2.11 Architektura MIDP



Copyright 1994-2007 Sun Microsystems, Inc.

Obrázek 3 – architektura MIDP

Jak je z obrázku patrné, kromě MIDP implementace může výrobce zařízení dodávat také rozšiřující API (OEM-Specific Classes). Aplikace, které tyto třídy využívají, jsou však závislé na daném zařízení, pro které byly vytvořeny. Například Siemens dodává k mobilním telefonům SL45i také API, které umožňuje posílat krátké textové zprávy SMS (což MIDP samo o sobě nenabízí). Na jedné straně je to výhodné, protože toto API umožňuje využít funkce telefonu, které by normálně MIDP API neumělo obsloužit. Na druhé straně však takto vytvořené aplikace nepoběží na jiném telefonním přístroji, než od firmy Siemens, což je trochu v rozporu s filozofií jazyka Java. Proto již dnes vznikají určitá standardizovaná API, na kterých pracují společně přední výrobci mobilních telefonů (například Game API pro snadné vytváření MIDP her).

### 2.11.1 MIDP 1.0

Mobile Information Device Profile (MIDP 1.0) je postaven na CLDC, přičemž implementuje rozhraní pro práci s životním cyklem aplikace, s HTTP protokolem, uživatelským rozhraním a persistentním úložištěm dat. MIDP 1.0 byl prvním dokončeným profilem pro J2ME, začal se tedy jako první implementovat a dnes se s ním můžeme například setkat u naprosté většiny mobilních telefonů podporujících J2ME.

Charakteristika zařízení s podporou MIDP:

- paměť použitelná pro prostředky Javy přibližně 512K (ROM + RAM),
- omezený výkon procesoru,
- omezený zdroj energie, typicky baterie,
- omezený prostor pro ukládání dat,
- připojení k bezdrátové síti (většinou s omezenou šířkou pásma),
- různorodost uživatelského interface, omezené ovládání (vstupní zařízení),
- malý display s nízkou hloubkou barev.

### 2.11.2 MIDP 2.0

MIDP 2.0 je nástupcem a rozšířením verze 1.0. Zpětná kompatibilita je zachována (obsahuje stejné třídy jako MIDP 1.0), přičemž přidává některé novinky.

#### Novinky:

- Podpora HTTPS - ta je založena na CLDC třídě *javax.microedition.io* (Generic Connection Framework),
- Multimédia - třída *javax.microedition.media* obsahuje vlastní Media API pro práci s multimedií. Toto API je jakousi podmnožinou Mobile Media API (MMAPI), což je rozsáhlé API pro práci s multimedií a umožňuje např. přehrávání videa, video-streaming apod. Media API narozdíl od MMAPI, zahrnuje podporu pouze pro audio, nikoliv pro video či grafiku, a to z toho důvodu, že některá zařízení s omezenými systémovými



prostředky nejsou schopna využít všechny možnosti, jež MMAPi poskytuje,

- Uživatelské rozhraní - hlavní změny proběhly ve třídě *javax.microedition.lcdui* - prvky Form a Item. Jen některé pro ilustraci - je možné lépe specifikovat pozici prvku ve formuláři, minimální velikost prvku, bylo rozšířeno ošetření uživatelských vstupů, možnost vytvářet vlastní Itemy (CustomItem),
- Game API - *javax.microedition.lcdui.game* - Game API poskytuje množství tříd, které umožňují snadněji vyvíjet hry pro bezdrátová zařízení. API snižuje množství kódu, které je nutné pro vytvoření aplikace, a tím také její velikost, která je jedním z hlavních limitujících faktorů. API používá standardní nízkoúrovňové grafické třídy z MIDPu (Graphics, Image apod.), a tak je v tomto API možné využívat metod těchto základních grafických objektů. Např. je možné vykreslit kompletní pozadí s využitím tohoto Game API a pak pomocí výše zmíněných základních tříd na něj kreslit základní grafické objekty jako jsou přímky, čtverce apod,
- Práce s RGB obrázky - souvisí s vývojem mobilních zařízení, pro něž je MIDP určen, a s grafikou obecně,
- Ověřování důvěryhodnosti MIDletu - na základě uživatelského nastavení bezpečnosti může MIDP odmítnout připojení na neověřené servery a podobně,
- Push architektura - s příchodem MIDP 2.0 se objevuje podpora tzv. PushRegistry. Je to podpora server-push modelu, kdy MIDlety mohou být aktivovány, jestliže obdrží příslušnou informaci od serveru. Následně se rozhodne podle nastavení aplikace, což je průnik nastavení samotného telefonu a uživatelského nastavení, jak daný MIDlet spustit: jestli se dotázat uživatele, spustit bez vyzvání či např. čekat na ukončení jiného běžícího MIDletu. Standardizovaná komunikace přes sériový port,
- Sdílené úložiště dat mezi MIDlety - data MIDletu jsou ukládána v tzv. RMS (Record Management System), což je persistentní úložiště. MIDP 1.0 neumožňovalo MIDletu číst data jiného MIDletu, jedná se tedy o velmi významnou inovaci,

- O-T-A (Over The Air) - Specifikace 2.0 nyní oproti verzi 1.0 vyžaduje podporu O-T-A, což je schopnost MIDPu instalovat, mazat, případně updatovat aplikace přes síť. Umožňuje rovněž poskytovateli aplikace zjistit, zda-li je daná aplikace na daném zařízení spustitelná a v neposlední řadě také získávat zprávy o stavu instalace či updatu dané aplikace na mobilním zařízení.

## **2.12 Aplikace**

MIDP aplikace se nazývá MIDlet. MIDlet je vždy rozšířením (potomkem) MIDlet třídy, což umožňuje application management softwaru kontrolovat běh aplikace, pracovat s jejími atributy a komunikovat s ní (aplikaci lze díky MIDlet třídě mimo jiné spouštět, pozastavovat, restartovat a vypínat). Tato třída tedy zprostředkovává rozhraní mezi aplikací a runtime prostředím. Application management software může spravovat více MIDletů. MIDlety jsou pak nazývány MIDlet suite. Fyzicky se aplikace sestává ze dvou částí - JAR a JAD soubor.

### **2.12.1 Java Archive (JAR)**

Aplikace je obsažena v tzv. balíčku (MIDlet Package). V něm jsou obsaženy všechny třídy, obrázky a další součásti, které aplikace používá. JAR soubor obsahuje i informace o MIDletu, které jsou potřebné ke spuštění aplikace - atributy, jako například název MIDletu, jméno autora a podobně.

### **2.12.2 Java Application Descriptor (JAD)**

Tento soubor obsahuje informace o aplikaci. Jde o obyčejný textový soubor, ve kterém jsou uvedeny atributy a jejich hodnoty (vždy "atribut: hodnota"), většinou stejné jako v JAR souboru.

## 2.13 Princip fungování J2ME

Aplikace pro J2ME se tvoří obdobně jako applety. V každém MIDletu musí být obsaženy tři základní metody, které určují start, pauzu a ukončení aplikace. Jednotlivé obrazovky jsou tvořeny speciálními metodami, které určují jejich obsah. Pohyb mezi nimi je umožněn prostřednictvím softwarových tlačítek, které jsou obvykle znázorněny ve spodní části obrazovky. K samotné obsluze je potřeba implementovat speciální rozhraní, které má za úkol sledovat stisknutá tlačítka a podle přání programátora provádět další akce.

(Graphic User Interface) MIDletů je rozděleno na dvě části: na vysokoúrovňové a nízkoúrovňové. Vysokoúrovňové GUI je tvořeno standartními prvky, jako jsou například textová pole, obrázky, indikátor průběhu, apod. Naopak nízkoúrovňové rozhraní umožňuje tvořit téměř libovolné tvary na takzvaném plátně (Canvas). Tohoto rozhraní využívají právě oblíbené Java hry, které takto mohou simulovat i 3D prostor. Je jen na schopnostech programátora, jak si umí s touto technologií poradit. Obecně doporučený postup je nejdříve si zjistit velikost obrazovky a poté vykreslovat jednotlivé tvary. To zaručí přenositelnost mezi různými přístroji s odlišnou velikostí displeje.

## 2.14 Vývojové prostředí

Standardní vývojové prostředí dodávané Sunem se jmenuje Java Wireless Toolkit. Toto prostředí obsahuje následující programy: KToolbar, emulátor, konvertor PRC souborů a preverify.

Začneme stěžejním programem KToolBar, který práci velmi usnadňuje. Jeho alternativou je použití manuálního zadávání příkazů, nebo technologie Ant. Tento program umí budoucí aplikaci spustit, zkompileovat i preverifikovat a následně zabalit do JAR souboru. MIDlet musí být totiž předběžně ověřen, protože tuto funkci, kterou jinak vykonává virtuální stroj Javy u stolních počítačů, mobilní přístroje neobsahují kvůli nedostatku místa. O tuto proceduru se stará právě preverify.

Chceme-li používat program i pro operační systém Palm OS (obecně PDA), musíme jej před distribucí zkonvertovat do tzv. PRC souboru, který bude posléze stažen do přístroje. Nic víc dělat nemusíte, tzn. žádné speciální upravování kódu pro Palm! To je právě velká výhoda přenositelnosti Javy.

Posledním programem je emulátor. Jak sám název napovídá, umí simulovat mobilní telefony a to buď černobílé nebo barevné. Dále je zde obsažen telefon značky Motorola i85s a RIMJavaHandheld, což je jakési PDA. Můžete si též vyrobit vcelku jednoduchým způsobem svůj vlastní mobil s požadovanou velikostí displeje a dalšími vlastnostmi, které mu nastavíte. Stačí mít obrázek telefonu a spoustu trpělivosti.

### 3. Zařízení používající J2ME

**3.1 Set-top-box:** MHP je zkratka pro Multimedia Home Platform, je to řešení na všechny problémy, které digitální televize v současnosti má. V podstatě potřebujete doma pouze jeden přijímač, jeden set-top-box, který bude schopný přijímat MHP od každého broadcastera. MHP je otevřený standard, je veřejně publikovaný, je přístupný každému a založený na programovacím jazyku Java. Společnosti Sony, Philips, Panasonic, Nokia, Grundig a Matsushita byly první, které v podstatě specifikaci MHP dostaly od začátku až tam, kde je dnes. MHP je nejen aplikačně programovací rozhraní. Pokrývá i aplikace security, transport, zpětný kanál k broadcasterovi, kontrolu životního cyklu aplikací atd. Platforma MHP pokrývá tři profily. Nejjednodušší (enhance) je v podstatě jen obohacená televize, digitální teletext. Další profil je interaktivita, kdy máte k dispozici zpětný kanál a můžete interaktivně komunikovat s broadcasterem, tzn. můžete se zapojit do různých her, můžete obchodovat přes televizní obrazovku atd. Poslední profil pokrývá Internet, web server přes váš televizor. Typické aplikace pro profil enhanced jsou počasí, zprávy, finanční zprávy, recepty. Když se díváte na TV program, kde se vaří, tak se vám objeví recept, případně si ho můžete nechat poslat domů e-mailem. Obohacený profil je dále rozšířený do interaktivního profilu přes zpětný kanál. Tzn., že se můžete účastnit různých znalostních her, voleb atd., používat home banking, e-mail a SMS. Poslední profil umožňuje brouzdat po Internetu. Uvedme v krátkosti některé konkrétní aplikace. Například ze Španělska počasí, cestování a Formule 1 v Německu, ve Finsku SMS, interaktivní hru z Belgie. Ale kdo je vlastně vytváří a bude vytvářet? Programátoři v Java jsou velmi drazí. Samozřejmě existují i automatizované nástroje, podobně jako MS Power Point, které aplikace vygenerují. Nedokážete s tím vytvořit každou aplikaci, ale aplikaci s počasím apod. vždy. Třetí nástroj – templates base, nabízí předdefinované šablony. Celý vtip interaktivních služeb MHP je vlastně založený na tom, že musíte vložit vaše soubory, vaše obrázky a vaše programy v Java do digitálního vysílání. Vaše aplikace přijde do karuselu, což je v podstatě jednoduchý videorekordér. Výstup může být multiplexovaný se signálem audio, video a opět je dekodovaný v TV přijímači. Samotný karusel nestačí na to, abyste mohli vysílat interaktivní služby. Potřebujete ještě nějaký servis-manager a řízení. Autoři aplikací nemusí být vždy

broadcasteri. Může to být např. i pražské letiště, které pošle svoji aplikaci přílety a odlety letadel, počasí na letišti atd. do České televize, která mu vloží tyto informace do vysílání přes multiplexor spolu s audiem, videem a diváci je mohou sledovat na svém digitálním přijímači.

**3.2 Mobilní telefony:** Jedna z oblastí Javy, která v posledních letech udělala velký pokrok jsou aplikace pro mobilní telefony. Odhaduje se, že celkový počet mobilních telefonů se pohybuje kolem jedné třetiny celosvětové populace, což je asi 6,5 miliardy lidí. Spousta her, vědecká kalkulačka, encyklopedie a e-mailový klient, to jsou jen první ukázky toho, co můžete s J2ME v mobilu získat. Java je totiž schopná kromě přístupu k e-mailu pomocí POP3 a SMTP nabídnout ještě mnohem víc. S Javou můžete přistupovat přímo k Internetu jako takovému, a to jak přes klasické CSD spojení, tak přes GPRS. Na rozdíl od prvních pokusů o Internet v mobilu jako například MS Mobile Explorer nabízí J2ME mnohem více, podporuje totiž třeba i SSL a některé další formy zabezpečených datových přenosů. LBS neboli Location Based Services (služby založené na vaší aktuální poloze), to je jen další z mnoha silných zbraní J2ME. Ta totiž dokáže pomocí vhodného programu zjistit vaší aktuální polohu z vašeho telefonu a nadále s ní pracovat.

Jedna z největších výhod J2ME je možnost stahování (popřípadě nahrávání) stále nových her do telefonu. To znamená, že mobily již nejsou vybaveny standardními hrami pro ten nebo onen model jako tomu bylo v minulosti, kdy například Nokia 3310 měla čtyři hry a pokud přestaly majitele telefonu bavit, měl smůlu, a mohl tak akorát vrhat závistivé pohledy na kamarády, vlastníci jiný model s jinými hrami. Nyní má telefon určitou velikost interní paměti (dříve v řádech kilobytů, dnes se již počítá ve stovkách megabytů), do té může majitel telefonu nahrávat nové hry až do té doby, než se paměť celá zaplní. Problémy s nedostatkem paměti můžete řešit vymazáním her, které již nehrajete. Toto je asi ta největší výhoda Javy. Ale nejenže můžete nabídku svých her neustále obnovovat a doplňovat, tyto hry se svou kvalitou a obsáhlostí nedají srovnat s těmi prvními hrami pro mobily, jako byl Snake.

Kvalita technického zpracování si nic nezadá se starými dobrými 16-ti bitovými konzolemi nebo počítači jako byla Amiga, Commodore atd. Některé by dokonce snesly srovnání i s hrami určenými pro počítače 386. Java hry mají,

na to jak jsou velké a kde se dají všude hrát, výbornou grafiku, občas i nějaké ty zvuky nebo dokonce jednodušší polyfonní hudbu. Sice to nejsou plnohodnotné herní tituly, se kterými se setkáváme třeba na PC nebo dnešních herních konzolách, ale rozhodně dokáží zabavit. Můžete je hrát prakticky kdekoli a stačí vám pouze váš mobil - právě v tomto je jejich síla. Obsáhlост současné nabídky herních titulů pro J2ME je velmi široká, ale to není příliš uspokojivé tvrzení, takže trochu podrobněji: dnes již existují stovky her určené pouze pro jeden model telefonu (např.: pro Nokii 3510i) a tisíce her jsou dostupné pro Mobilní Javu jako takovou, ty ovšem nemusí být kompatibilní se všemi výrobci a všemi jejich modely.

Java hry určitě nenabídnou takovou skvělou 3D grafiku, takové zvuky a hudbu, jako hry pro N-GAGE a její následovníky, ale zamyslete se: jedna hra pro N-GAGE stojí 1500-2500 Kč. Jedna Java hra stojí 30 Kč plus poplatky za připojení k WAPU, přes který ji musíte stáhnout. A ještě je tu jedna věc mluvící pro Javu – majitelé datových kabelů. Ti si Java hru stáhnou z internetu a do mobilu nahrají z počítače. Koncová cena jedné Java hry potom vyjde prakticky na 0 Kč. Myslím, že hry pro J2ME se budou hrát tak dlouho, dokud nepřijde nějaká nová, ale obdobně fungující technologie.

Ale na druhou stranu vývojové týmy, které budou chtít vydělat více peněz, budou raději dělat hry pro N-GAGE, už jenom proto, že J2ME nenabízí žádný způsob protipirátské ochrany, takže porušování autorských práv je v této oblasti řekněme „denním chlebem“ obrovského množství lidí.

**3.3 Smartphone** je označení pro tzv. chytrý telefon, tj. telefon s nějakým otevřeným operačním systémem (např. Windows Mobile, Symbian). Zařízení Smartphone se systémem Microsoft Windows nebo Symbian otevírá novou kapitolu technologie mobilních telefonů. Slučuje v sobě to nejlepší z mobilní hlasové a datové komunikace s nejlepšími funkcemi softwaru pro správu osobních informací. Zařízení Smartphone disponuje bohatou nabídkou aplikací, vysoce integrovanými telefonními funkcemi a otevřenou platformou pro vývoj. Mobilní telefony na bázi platformy Smartphone nabízejí hlasovou i textovou komunikaci a bohatou nabídku aplikací pro bezdrátovou práci s daty. Výsledkem je osobní a skutečně výkonný mobilní telefon pro práci i volný čas.

Zařízení Smartphone je podobně jako Pocket PC vybaveno systémem Microsoft Windows CE, operačním systémem pro práci v reálném čase. Zařízení Smartphone je optimalizováno s důrazem na co nejlepší hlasovou a textovou komunikaci a vychází vstříc požadavkům zákazníků na cenu, velikost, výkonnostní parametry a estetický design.

Základní výhodou je rozšiřitelnost, kterou klasické mobilní telefony postrádají, neboť jak pro Symbian tak pro Windows Mobile existuje velké množství aplikací z různých odvětví, kterými lze telefon rozšířit (například kvalitnější internetový prohlížeč, mobilní verze balíku Office, kvalitní emailový klient, souborový manažer, multimediální přehrávač a mnoho dalších), pak také možnosti personalizace a zobrazení, podpora paměťových karet a v neposlední řadě velmi jednoduchá synchronizace s počítačem.

Mezi největší nevýhody smartphone mnohdy patří větší rozměry samotného přístroje (které však pro lidi s většími prsty nejsou často vůbec na škodu), menší výdrž na baterie (ta může být u některých chytrých telefonů, například Motorola MPx200, při běžném používání i cca 1-2 dny), a v některých případech cena, která však bývá úměrná tomu, co dané přístroje ve skutečnosti dokáží.

V poslední době se chytré telefony dostaly do podvědomí i mnoha běžným uživatelům a to především díky výraznému snížení jejich cen: například Motorola MPx220 nebo Nokie 3230 se prodávají za cenu okolo pěti až sedmi tisíc korun.

Operační systém Windows Mobile v základu Java aplikace vůbec nepodporuje, naštěstí existují aplikace třetích stran, díky kterým si podporu Javy snadno doinstalujete. Mezi nejlepší a nejpoužívanější emulátory patří bezesporu IBM DELTA MIDP 2.0, který zvládá i podporu Javy verze 2.0, ale bez problémů na něm spustíte i starší midlety pro javu 1.0. Dále pak můžete vyzkoušet emulátory JavaMAN, Java Virtual Machine (by VXD). Bohužel některé midlety v emulátoru nefungují, musíte tedy vyzkoušet jaké java soubory (rozumějte pro jaké verze telefonů) emulátor zvládá dobře a tím se pak také řídit.



**3.4 PDA:** Pod pojmem PDA (Portable Digital Assistant) je souhrnně označována třída „kapesních“ počítačů. Tyto počítače se vyznačují dotykovým displejem a malými rozměry. Tato zařízení většinou nemají klasickou klávesnici (i když k některým zařízením ji lze pořídit). K ovládání využívají právě dotykový displej. Nedílnou součástí těchto zařízení je rozpoznávání písma. V dnešní době jsou na trhu s těmito zařízeními dvě dominantní platformy: Palm a PocketPC.

#### **3.4.1 Palm**

Firma Palm vyrábí nejen vlastní zařízení, ale také dodává pro tato zařízení programy, hlavně operační systém označený obchodním názvem PalmOS. Svoji činnost zahájili na podzim roku 1996, kdy firma Palm představila první verzi operačního systému a první zařízení, které bychom dle dnešních kritérií označili jako PDA. Kapesní počítače Palm obsahují procesory mnoha výrobců např. Motorola, Intel, Samsung atd. Aktuálně tyto čipy pracují na frekvencích řádově stovek MHz. Interní paměť se pohybuje od 32 MB. Nejnovější modely obsahují sloty pro přídavné paměťové karty a konektory pro připojení hardwarového rozšíření (např. modul GPS). Standardně je také zajištěna konektivita a možnost synchronizace s PC.

#### **3.4.2 Pocket PC**

Rok po uvedení prvního přístroje firmy Palm, představil Microsoft nový operační systém Windows CE. Tento produkt vychází z tradičního operačního systému této firmy Windows. Hlavními požadavky při vývoji systému byly menší paměťové nároky a grafické rozhraní přizpůsobené pro zobrazovací zařízení s poměrně malým rozlišením. Zařízení určená pro provoz Windows CE verze 3.0 a vyšší se nazývají Pocket PC. Mezi největší výrobce Pocket PC dnes patří firmy Compaq, Hewlett-Packard, NEC a Toshiba. Zařízení jsou osazována procesory pracujícími na frekvencích řádově ve stovkách MHz. Interní paměť se pohybuje od 32 do 128 megabajtů. Displeje jsou barevné s rozlišením až 352 x 288 pixelů. I tato třída zařízení většinou podporuje přídavné paměťové karty a další hardwarová rozšíření, stejně tak nabízí možnost připojení a synchronizace s osobním počítačem.

PDA má mnohem větší možnosti, než většina ostatních mobilních zařízení jako třeba mobilní telefony. Jde zejména o velikost paměti, rychlost zpracování instrukcí a v neposlední řadě také velikost zobrazovací plochy. První, co je třeba při tvorbě aplikace provést, je zvolit prostředí (platformu), na níž bude celá aplikace stát. Volba prostředí bude záviset na operačním systému PDA. Výše v sekci použitých technologií bylo zmíněno, že existují dva druhy PDA zařízení. Palm a Pocket PC. Každý z nich má jiný operační systém a tím i jinou podporu zmíněných platforem.

S podporou Javy je to u Pocket PC složité. Standardně v podstatě žádná podpora není. Pokud na zařízení má být provozován Javovský kód, je nutné nejprve obstarat virtuální stroj a ten do zařízení nainstalovat. Na trhu je mnoho druhů JVM pro Pocket PC, některé komerční, některé zdarma. Obvykle však ty bez poplatku podporují starší verze Javy. Nicméně je možné na Internetu najít vhodný virtuální stroj a do zařízení jej nainstalovat.

Tyto praktické aplikace použití mobilní Javy jsem shledal jako stěžejní. Další zařízení již pracují mnou popsanými principy. Patří sem např.: pagery, videotelefony apod.

## **4. Platformové problémy a jejich řešení**

Při vývoji J2ME stáli vývojáři před velkým problémem. Jak vmístit přenositelnou verzi JAVY do přístroje, který má minimální prostředky. Nejen, že zařízení, pro která je J2ME určena, mají malý display a téměř žádnou klávesnici, ale hlavně mají minimální úložný prostor a pomalý procesor. S tímto problémem se SUN vypořádal vypuštěním nepotřebných tříd a balíčků a omezením těch zbylých. V případě CLDC navíc ještě omezením virtuálního stroje.

### **4.1 J2ME a bezpečnost**

Výrazným omezením v rozletu javových aplikací je bezpečnostní model aplikace v mobilním telefonu. Společnost Sun Microsystems viděla problém v aplikacích, které bez uživatelského vědomí provádějí nebezpečné věci se seznamem kontaktů, rozesílají tajně SMS a ovlivňují jiné javové aplikace. Proto byl pro Javu zvolen bezpečnostní model nazvaný Sandbox, díky kterému nemůže aplikace spouštět jiné aplikace a nemá přístup ke zdrojům mobilního telefonu jako jsou telefonní seznam, SMS, obrázky nebo melodie. Aplikace také nemá přístup k datům uloženým jinou aplikací ani ke kartě SIM a nemůže iniciovat hlasové spojení.

Díky těmto vlastnostem musíme zapomenout na různé aplikace vylepšující práci s kalendářem, adresářem a SMS. Některé telefony sice mají výrobcem doplněné knihovny pro tento přístup, ovšem jedná se o nestandardní postup, který zamezuje přenositelnost aplikací. Na druhou stranu je prostředí zajišťující bezpečnost výhodné pro aplikace s kritickými požadavky na bezpečnost, například bankovní. Zde nám naopak vyhovuje, že jiná aplikace nemůže přechít uživatelské heslo ani šifrovací klíče uložené v aplikaci.

Spolu s možností jednoduché instalace javových programů kráčí ruku v ruce otázka bezpečnosti. Pro operační systém Symbian, který obsahuje čím dál více inteligentních telefonů, se již objevilo několik virů, i když bez aktivní pomoci uživatele by ani jeden z nich neměl šanci. A na podzim loňského roku také došlo k odhalení bezpečnostní mezery v VM.

#### 4.1.1 Koncept bezpečnosti

V jazyku Java je na bezpečnost kladen velký důraz. Z důvodů omezeného množství zdrojů na malém zařízení byl koncept bezpečnosti pro CLDC oproti standardní edici Javy trochu zjednodušen. Tento koncept zahrnuje tři úrovně - nízkoúrovňovou bezpečnost na úrovni virtuálního stroje, bezpečnost na úrovni aplikace a *end-to-end* bezpečnost.

#### 4.1.2 Nízkoúrovňová bezpečnost na úrovni virtuálního stroje

Tento typ bezpečnosti zaručuje, že budou spuštěny pouze javové třídy odpovídající sémantice jazyka Java, které nemohou svým během poškodit zařízení, na kterém aplikace běží. Javové třídy jsou po překladu do bajtkódu *preverifikovány* (předověřeny). Během tohoto procesu jsou do .class souborů přidány dodatečné atributy. Na zařízení probíhá na základě těchto atributů verifikace, která ověří, že aplikace nemůže poškodit paměť mimo rámec paměti alokované aplikací. Pokud jsou atributy preverifikace neplatné nebo chybí, je aplikace odmítnuta.

#### 4.1.3 Bezpečnost na úrovni aplikace

Javová aplikace může použít pouze ty knihovny a zdroje, které jí umožní aplikační prostředí použít. Základem je model *sandbox* (pískoviště) společný s applety. Aplikace běží v omezeném prostředí, jehož hranice nemůže překročit. Toto prostředí splňuje následující body:

- Soubory .class jsou správně verifikovány,
- Stažení, instalace a vůbec provoz javové aplikace zajišťuje standardní mechanismus, který programátor aplikace nemůže žádným způsobem ovlivnit,
- Aplikace má k dispozici pouze knihovny CLDC, MIDP a knihovny přidané navíc výrobcem zařízení,
- Sada nativních funkcí přístupná virtuálnímu stroji je uzavřená, aplikace nemůže nahrát žádnou další knihovnu rozšiřující přístup k nativním funkcím poskytnutým výrobcem zařízení.

Dále jsou chráněny systémové knihovny. Aplikace je nesmí předefinovat ani do nich přidávat nové třídy. Není možné si definovat vlastní *zavaděč třídy* (class loader). Aplikace může kromě knihoven používat pouze třídy ze svého JAR souboru a odnikud jinud.

#### **4.1.4 "End-to-end" bezpečnost**

Zahrnuje bezpečnost transakce mezi zařízením a například serverem umístěným v internetové síti. Tento typ bezpečnosti CLDC přímo nepopisuje, je součástí profilu MIDP 2.0, kde bylo přidáno HTTPS spojení, zajišťující tuto *end-to-end* bezpečnost. Profil MIDP 1.0 tuto úroveň bezpečnosti nezahrnuje.

#### **4.1.5 Důvěryhodné a nedůvěryhodné sady midletů v MIDP 2.0**

Do specifikace MIDP 2.0 byl přidán mechanismus podpisování aplikací - aplikace mohou být nepodepsané, což znamená, že jsou *nedůvěryhodné*, nebo mohou být podepsané, tedy jsou *důvěryhodné*. S MIDP 1.0 aplikacemi se na zařízení podporujícím MIDP 2.0 zachází jako s nedůvěryhodnými aplikacemi. Rozdíl mezi důvěryhodnými a nedůvěryhodnými sadami midletů spočívá v tom, zda a za jakých okolností je jim povolen přístup k rozhraním, jež jsou považována za citlivá.

## 4.2 Hardwarové omezení J2ME

Každý výrobce mobilních telefonů má jiné představy o ovládání i ergonomii, a tak se i přes standardizované uživatelské rozhraní na různých telefonech javovská aplikace jinak ovládá a často i jinak vypadá. Vytvořit přenositelnou a ovladatelnou aplikaci pro co největší počet mobilních telefonů je značně obtížný úkol. Nakonec i samotný standard J2ME MIDP 2.0 je různými výrobci interpretován tak, že se přenositelnost aplikace stává pro vývojáře noční můrou. Problémy s kódováním, následují problémy s maximálními bloky použité paměti, fragmentací paměti a odlišnostmi v komunikačních algoritmech. Při znalostech uvedených rysů mobilní Javy se nabízí otázka, zda se opravdu jedná o platformu vhodnou pro vývoj mobilních aplikací. Odpověď zní ano, pokud se ovšem dokážeme vypořádat s jejími omezeními.

Jedno ze zásadních omezení, vyplívajících z rozměrů přístroje, je velikost napájecího zdroje. V této oblasti dochází v poslední době ke značnému posunu vpřed. V úvahu přichází technologie tzv. superkondenzátorů. Velké využití těchto "malých baterií" může v budoucnu pomoci také kategorii mobilních zařízení, kde s rostoucí integrací obvodů klesá i napájecí napětí i spotřeba. Výdrž by sice byla nižší než u chemických akumulátorů, ale schopnost neuvěřitelně rychlého nabití v řádu sekund je lákavá. Jako příklad poslouží představa nabití mp3 přehrávače za několik sekund a pak jeho provoz po několik hodin.

Dalším omezením je energetická náročnost zobrazovací jednotky. K řešení problémů s napájením displeje, nepochybně přispěje použití technologie OLED, jež využívá organických elektroluminiscenčních diod. Pochází z roku 1987, kdy jí vyvinula firma Eastman Kodak. Technologie fosforeskujících OLED dosahuje 4x větší účinnosti než „normální“ OLED technologie. Využívá principu elektrické fosforescence, která převádí až 100 % elektrické energie na světlo. To je, v porovnání s účinností 25-30 % u „klasických“ OLED a jen cca 10 % u LCD obrazovek, velký pokrok. Při jasu 200 cd/m<sup>2</sup> dosahuje spotřeby pouze 125 mW, tedy výrazně méně než podsvětlené LCD (240mW). Nejnovější PHOLED jsou při napětí 6,5 V schopny dosáhnout osvětlení 18 lm/W a jasu 1000 cd/m<sup>2</sup>. Tedy výrazně větší jas než nejnovější LCD s 600 cd/m<sup>2</sup>.

### 4.3 Softwarová omezení J2ME

Velikost displeje se různí nejen různými výrobci, ale dokonce také jednotlivými typy zařízení. Je třeba počítat se zcela jiným grafickým uživatelským rozhraním u verze J2ME než J2SE. Proto je třeba optimalizovat softwaru. Grafické uživatelské rozhraní je definováno v MIDP. To znamená, že grafické prvky jsou zahrnuty v profilech a ne v konfiguracích. Uživatelské rozhraní MIDP se rozděluje na API: vysokoúrovňové (*high-level*) a nízkoúrovňové (*low-level*).

Vysokoúrovňové API bylo vytvořeno pro aplikace, které mají být přenositelné mezi více mobilními zařízeními. Používá vyšší úroveň abstrakce a poskytuje velmi malý prostor pro úpravy. Toto se projevuje následovně:

- Aktuální kreslení na displej je závislé na způsobu implementace. Uživatel si nemůže sám zvolit vzhled (např. barvu, podobu, fonty, atd.) komponent,
- Interakce s komponentami je zapouzdřena v implementaci MIDP a aplikace se o toto působení nestará,
- Aplikace nemůže poskytovat konkrétní vstupní ovládání jako jsou například speciální tlačítka na mobilním zařízení.

Nízkoúrovňové API používá menší úroveň abstrakce. Toto API je navrženo pro aplikace, které potřebují přesné umístění a kontrolu nad grafickými prvky a potřebují mít přístup k nízkoúrovňovým událostem (*low-level events*). Příkladem takových aplikací jsou třeba hry. Použití nízkoúrovňových API v aplikacích může:

- Mít plnou kontrolu nad tím, co právě má být vykresleno na displeji. To znamená, že si můžeme definovat barvu, fonty a různé tvary, které chceme vykreslit,
- Reagovat na události jako jsou stisknutí či uvolnění tlačítka.

## 4.4 Omezení virtuálního stroje pro CLDC

Virtuální stroj Javy má z důvodu úspory paměťového prostoru a snížení zátěže procesoru tato hlavní omezení:

- nepodporuje plovoucí řádovou čárku<sup>4</sup>,
- nelze použít finalizaci<sup>5</sup>,
- omezení zpracování chyb<sup>6</sup>,
- JNI (Java Native Interface) není k dispozici<sup>7</sup>,
- neumožňuje zavedení zavaděče tříd<sup>8</sup>,
- chybí skupiny vláken<sup>9</sup>,
- zabezpečení na úrovni virtuálního stroje.<sup>10</sup>

---

<sup>4</sup> *Nepodporována plovoucí čárka* – aplikace nemohou používat datové typy jako double nebo float. Důvodem tohoto omezení je zejména to, že desetinnou čárku nemusí podporovat zařízení na úrovni hardwaru.

<sup>5</sup> *Není finalizace* – neumožňuje provést úklidové práce s objekty (např. uzavření zdrojů) před vlastním zrušením objektu pomocí garbage collectoru.

<sup>6</sup> *Omezené zpracování chyb* – zpracování chyb je závislé na implementaci. Chyby nesouvisející s během programu, jsou zpracovány podle možností daného zařízení.

<sup>7</sup> *JNI* – aplikaci z bezpečnostních důvodů není umožněn přístup k funkcím poskytovaným operačním systémem. Je možné přistupovat pouze k funkcím poskytovaným rozhraním Java konfigurace CLDC nebo některého z jejích profilů.

<sup>8</sup> *Nejsou zavaděče tříd* – KVM má vestavěn zavaděč tříd. Ten uživatel nemůže nahradit ani změnit.

<sup>9</sup> *Chybí skupiny vláken* – pro provádění operací se skupinami vláken je nutné v aplikaci vytvořit kolekci objektů.

<sup>10</sup> *Zabezpečení na úrovni virtuálního stroje* – stará se o ně verifikátor tříd (nesmí být odkazy na neplatná místa v paměti, zajištění práce odpovídající specifikaci KVM). Jsou dva stupně verifikace: preverifikace při vývoji aplikace, minimální verifikace v zařízení.



Požadavky MIDP na zařízení, jež definuje specifikace MIDP, jsou požadavky nad CLDC, platí současně s požadavky na CLDC.

Minimální požadavky MIDP na hardware jsou:

na display

- velikost 96x54,
- hloubka barev: 1-bit,
- poměr horizont. a vertikálních pixelů: přibližně 1:1.

na způsob vkládání dat

- vkládání dat alespoň jedním z těchto způsobů: ITU-T telefonní klávesnice, QWERTY klávesnice, dotykový displej.

na paměť

- 128 kB trvalé paměti pro MIDP komponenty (MIDP aplikaci jako takovou). Trvalou pamětí se rozumí paměť, která se po vypnutí přístroje nesmaže,
- 8 kB trvalé paměti pro data uložené MIDP aplikací,
- 32 kB dočasné runtime paměti (Java heap).

na síť

- obousměrný síťový provoz s omezenou rychlostí přenosu.

Tato omezení MIDS profilu by měla upravovat připravovaná veze 3.0. Hlavní vylepšení jsou popsána v následujícím textu.

## 4.5 MIDP 3.0

Třetí verze mobilní Javy by mohla způsobit opravdovou revoluci. Na nové verzi implementace mobilní Javy se již podle všech informací usilovně pracuje. Už je také na čase. Mobilní telefony jsou stále výkonnější a žádají si větší využití svých schopností. Přeci jen některá omezení současného profilu MIDP 2 jsou velmi limitující.

- podpora více spuštěných midletů současně,
- běh midletu na pozadí,
- možnost automatického spuštění (například po zapnutí telefonu),
- minimalizování rozdílů mezi jednotlivými platformami a tedy například zvýšená kompatibilita midletů mezi telefony různých značek (jeden z velkých problémů současné mobilní Javy),
- vylepšena podpora velkých displejů, případně možnost využívat i druhý displej,
- zlepšena přímá podpora mobilních her,
- samozřejmě také spolupráce mobilní javy s jinými funkcemi telefonu jako například MMS, SyncML, paměťová média atd,
- měla by být také vylepšena podpora lokalizace do různých jazyků.

## 5. Závěr

Cílem této práce bylo komplexně zdokumentovat, popsat a následně navrhnout možné řešení platformových problémů programovacího jazyka Java pro mobilní zařízení. V následujících řádcích se budu snažit vysvětlit, proč právě Java je tím správným nástrojem jak pro tvorbu aplikací tak i pro jejich správné provádění.

Širokému výčtu zařízení, jež J2ME využívají, dominuje oblast mobilních telefonů. Zde se Java uplatňuje nejvýrazněji. Velký boom dnes zažívají především hry, avšak vývoj se nezastavil také v uživatelském softwaru. Jako příklad si můžeme uvést různé emailové klienty, internetové prohlížeče a také třeba ICQ komunikátory. Za zmínku stojí možnost snadnému přístupu uživatele k informacím na internetu. Toto by mohlo v budoucnu kupříkladu přispět ke snížení dopravních komplikací na našich přetížených silnicích. Představme si, že bude existovat systém, jež bude motoristům průběžně posílat informace o vzniklých komplikacích a navádět je na objízdnou trasu po méně frekventovaných ulicích. To je ovšem zatím hudba budoucnosti. Podle mého názoru, je toto správný směr, jakým by se měl vývoj ubírat.

Vyvstává tu však otázka, zda Java ustojí překotný vývoj telefonů, případně PDA, které disponují vlastním operačním systémem a to ať již myslíme Windows Mobile či Symbian. Dle mého názoru J2ME o svůj post nepřijde. Hlavní důvod vidím v přenositelnosti softwaru na široký okruh mobilních zařízení a z toho plynoucí zisky pro jejich vývojové pracovníky. Speciální kapitolu tvoří hry u nichž je na prvním místě výkon procesoru a použitelná paměť.

Možnosti Javy dnes již nejsou omezeny telekomunikačními limity přenosových technologií. Ty nám poskytují možnost trvalé konektivity a dostatečné přenosové rychlosti pro jakoukoliv aplikaci napsanou v Javě. Jistým omezením může být ztráta konektivity, způsobená rychlým pohybem uživatele, např. jízda automobilem apod. To už však není problém samotné Javy, ale telekomunikačních společností.

Popsal jsem Javu a pokusil jsem se najít možná řešení jejích platformových problémů. V první části bakalářské práce uvádím stručnou historii a popis architektury jazyka včetně základních principů jeho fungování. V další části se zabývám dokumentací co možná nejširšího počtu zařízení, jež používají tuto technologii. Ve dvou posledních částech jsem popsali stávající problémy J2ME a navrhl možná řešení.

Z mého výkladu jasně vyplývá, že Java pro mobilní zařízení, je velmi silná platforma, jež bude v následujících letech sloužit jako nástroj pro vývoj aplikací a také pro práci s nimi.

.

## Seznam použité literatury:

1. JUNTAO Y.M.: *Enterprise J2ME: Developing Mobile Java Applications*, 448 p., Prentice Hall PTR; 1st edition (October 20, 2003)
2. TOPLEY K.: *J2ME in a Nutshell*, 478 p., O'Reilly Media, Inc.; 1 edition (March 20, 2002)
3. Farley J., Crawford W., Flanagan D.: *Java Enterprise in a Nutshell*. O'Reilly & Associates Inc., Sebastopol, 2002, ISBN 0-596-00152-5.
4. MUCHOW J.W.: *Core J2ME Technology*, 736 p., Prentice Hall PTR; 1st edition (December 21, 2001)
5. Chappell D., Jewell T., *Java Web Services*. O'Reilly & Associates Inc., Sebastopol, 2002, ISBN 0-596-00269-6.
6. Sun Microsystems, Inc.: J2ME application program interface  
<http://java.sun.com/j2me/>
7. Java ME-Wikipedia, otevřená encyklopedie  
<http://cs.wikipedia.org/wiki/j2me/>
8. Dioné: Programovací jazyk Java  
<http://dione.zcu.cz/java/>
9. Interval  
<http://interval.cz/clanky/j2me-a-bezpecnost/>
10. RDC  
Portal <http://www.rdc.cz/index.php?jazyk=0&sid=2&main=sekce&stav=clanek&cid=1178>.

## **Seznam obrázků**

- Obrázek 1    Přehled použití *Java 2 Platform*
- Obrázek 2    Architektura J2ME
- Obrázek 3    Architektura MIDP